

JS Level 2



ИНСТРУМЕНТЫ



Инструменты

Для прохождения основной части курса вам понадобятся два инструмента: Node.js и редактор VS Code (для дополнительных лекций понадобятся также другие инструменты).

Кроме того, важны следующие три момента:

1. У вас должны быть права администратора на компьютере (чтобы вы могли устанавливать программы);
2. Ваш пользователь должен называться по-английски, без пробелов в имени (если это не так – переименуйте);
3. Создавайте все проекты где-нибудь на диске **C:**, например, в каталоге **projects** (следите за тем, чтобы в именах каталогов и файлов не было пробелов, не английских символов и т.д.).

В любом случае, если у вас возникнут проблемы, пишите в канал курса.



Важно

Если у вас Windows 7, то придётся обновиться до Windows 10 (поскольку Node.js уже не поддерживают Windows 7), либо использовать обходные пути (см. слайд "Установка для Windows 7).



Node.js

Node.js – это специальная среда для запуска JS приложений, позволяющая запускаться им вне браузера.

На Node.js написано большинство инструментов, которые используются Frontend-разработчиками.



VS Code

VS Code – это редактор кода, т.е. приложение, в котором вы будете писать код. На сегодняшний день является самым популярным в среде веб-разработчиков.



Node.js для Windows 7

Выберите [Windows Binary \(.zip\)](#) для своей операционной системы (32 или 64 бит):

LTS Recommended For Most Users	Current Latest Features
 Windows Installer <small>node-v14.17.1-x86.msi</small>	 macOS Installer <small>node-v14.17.1.pkg</small>
 Source Code <small>node-v14.17.1.tar.gz</small>	

Windows Installer (.msi)

Windows Binary (.zip)

macOS Installer (.pkg)

macOS Binary (.tar.gz)

Linux Binaries (x64)

Linux Binaries (ARM)

Source Code

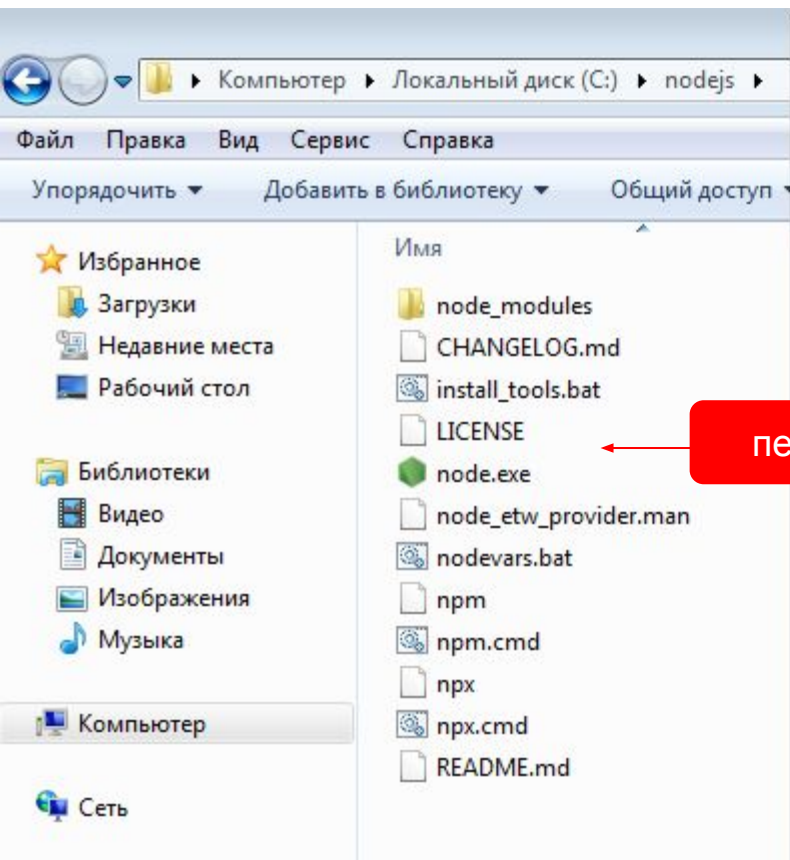
32-bit	64-bit
32-bit	64-bit
64-bit	
64-bit	
64-bit	
ARMv7	ARMv8
node-v14.17.1.tar.gz	



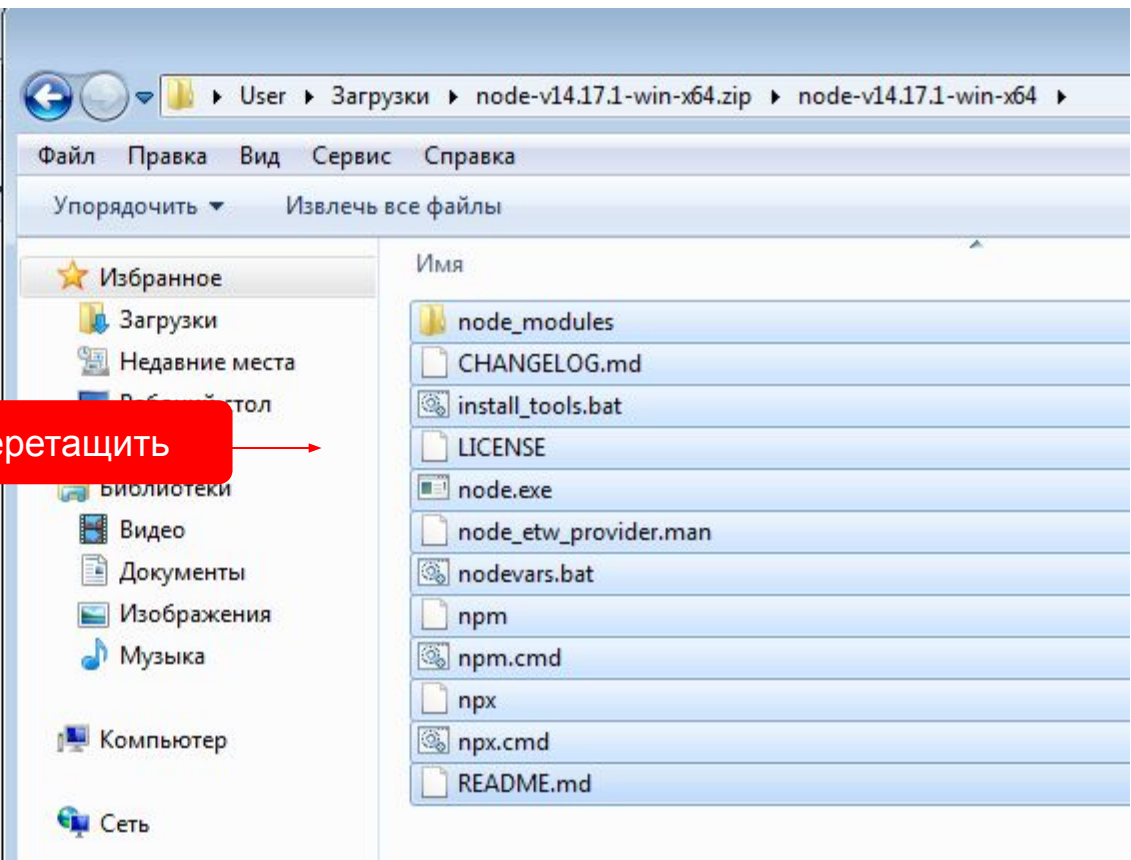
Node.js для Windows 7

Создайте на диске **C:** каталог **nodejs** и распакуйте туда архив так, чтобы файлы попали в сам каталог **nodejs**.

Каталог C:/nodejs



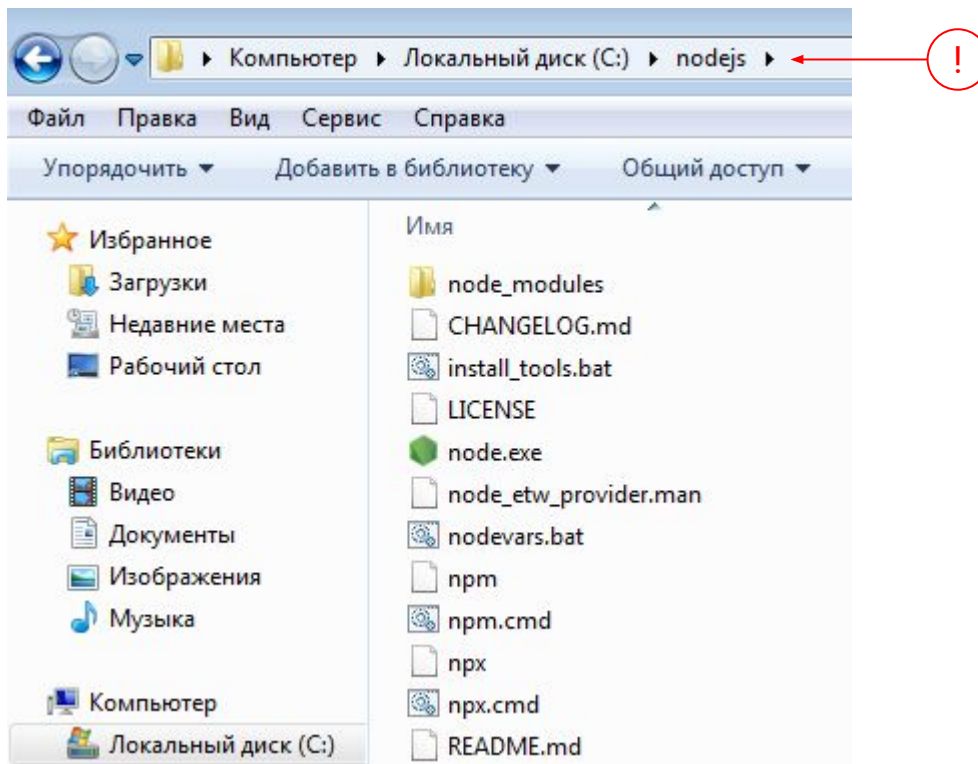
Скачанный каталог



перетащить

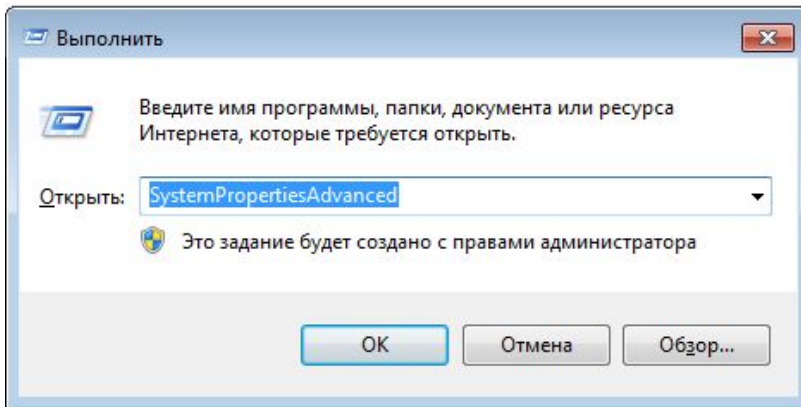
Node.js для Windows 7

Убедитесь, что именно файлы попали именно в `C:/nodejs`, а не во вложенный каталог:



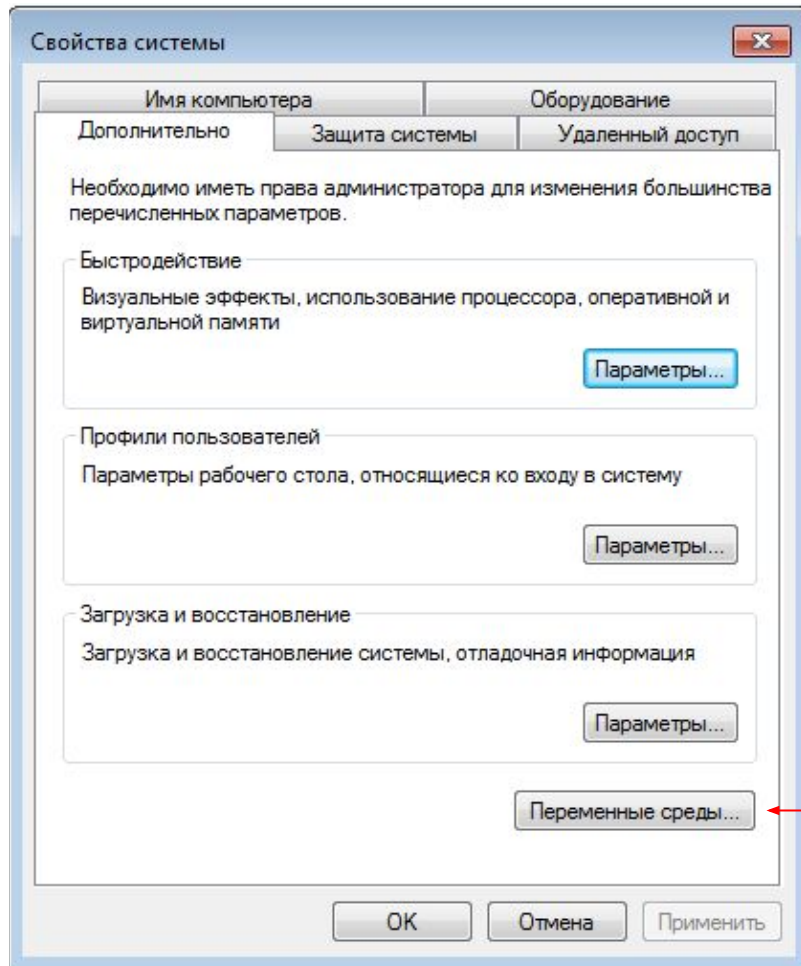
Node.js для Windows 7

Нажмите **Win + R**, введите **SystemPropertiesAdvanced** и нажмите **Enter**:



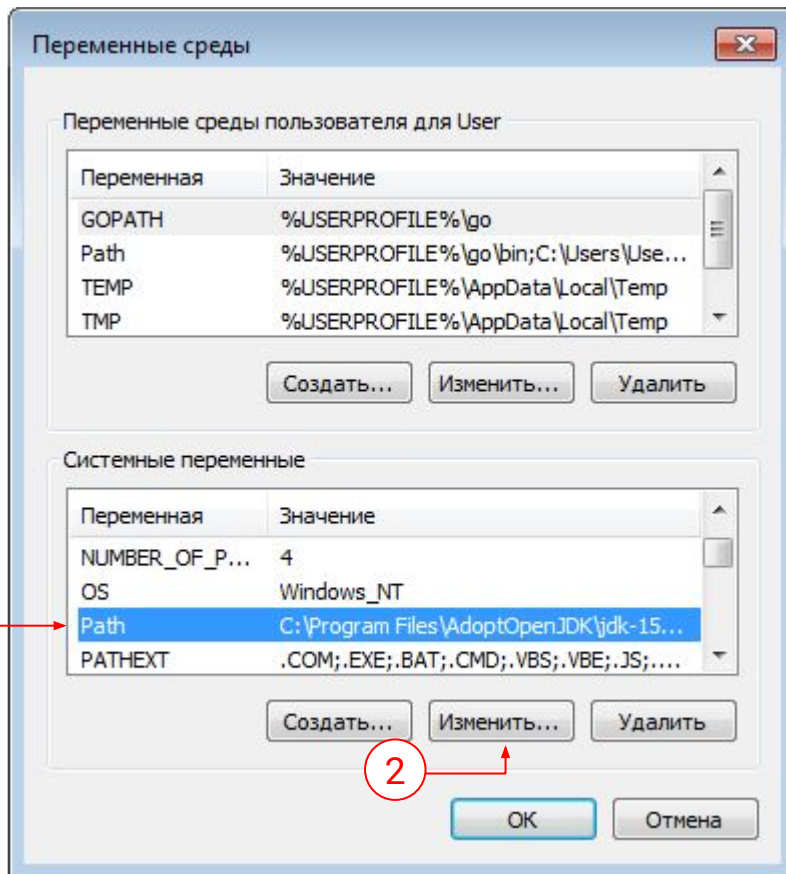
Node.js для Windows 7

Нажмите на кнопку **Переменные среды**:



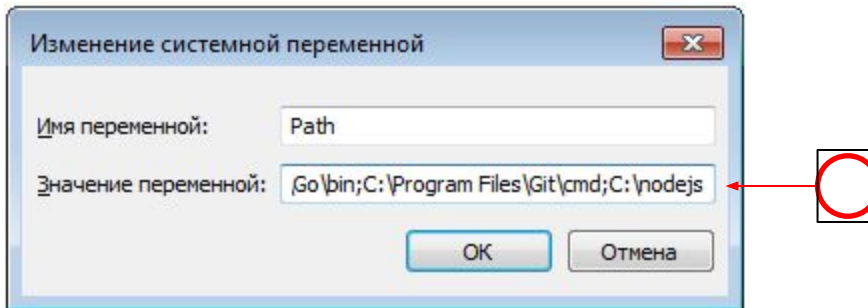
Node.js для Windows 7

В нижней части экрана (раздел Системные переменные) выберите **Path** (1) и нажмите на кнопку **Изменить** (2):



Node.js для Windows 7

С помощью стрелки вправо на клавиатуре аккуратно пролистайте до конца и допишите `;C:\nodejs`, именно так с `;` в начале, и нажмите на кнопку **OK**:

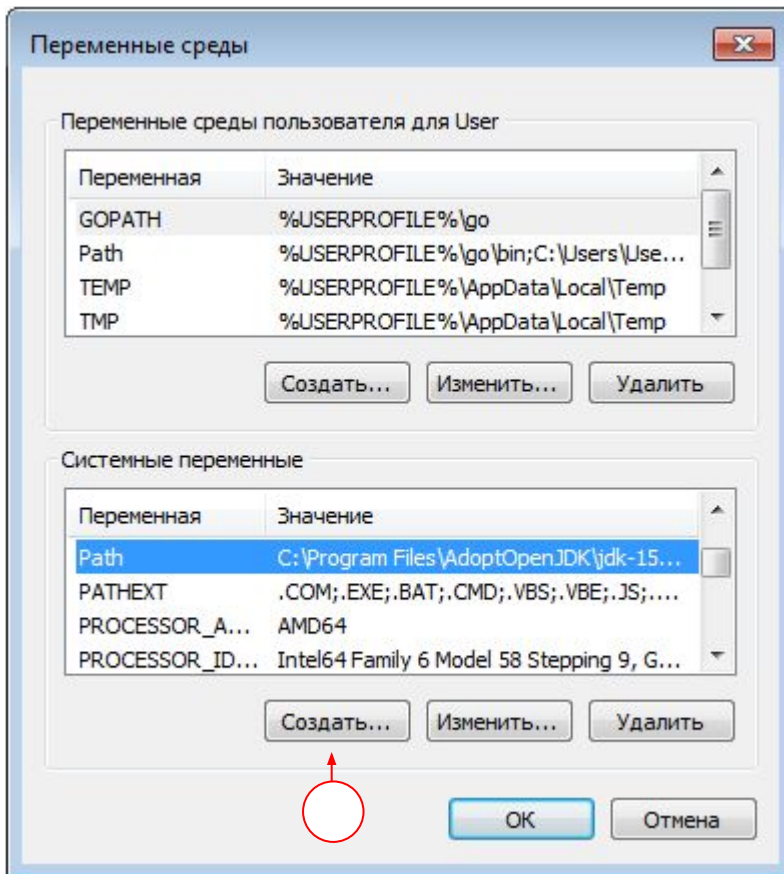


Если что-то пошло не так, нажмите на кнопку **Отмена** и повторите всё заново.



Node.js для Windows 7

После этого нажмите на кнопку **Создать** в предыдущем окне:

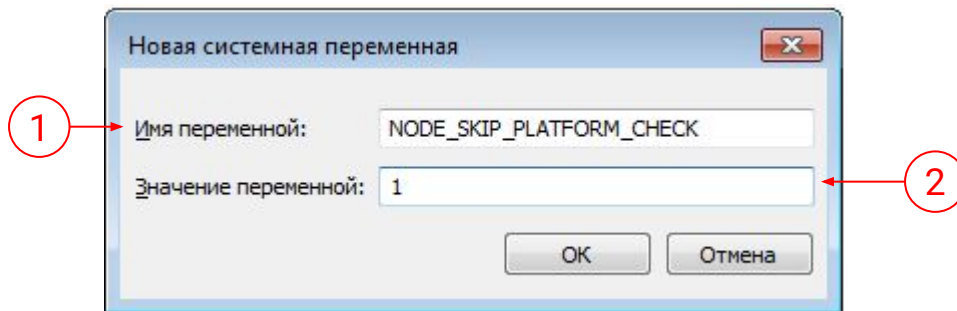


Node.js для Windows 7

Заполните следующими данными:

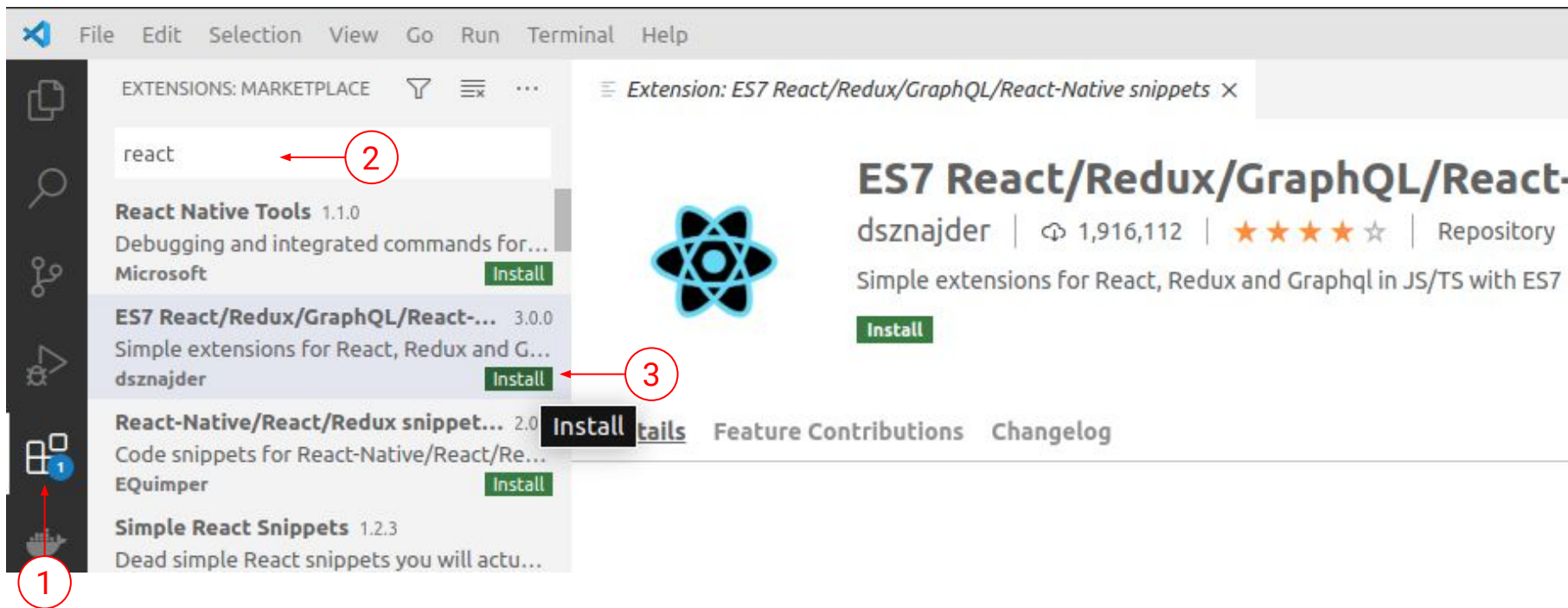
1. Имя переменной: `NODE_SKIP_PLATFORM_CHECK`;
2. Значение: `1`.

После чего нажмите на кнопку **OK** и закройте все открытые окна.



VS Code React Extension

Запустите VS Code, зайдите в панельку **Extensions (1)**, наберите **React (2)** и нажмите на кнопку **Install (3)**:



Убедитесь, что вы устанавливаете расширение от [dsnajder](#).



JS & WEB API



JS & Web API

Несмотря на то, что JS – это единственный язык, полноценно поддерживаемый браузерами, и, в первую очередь, используется для создания веб-приложений, работающих в браузере, сам JS не имеет никаких средств для этого.

Что это значит? Это значит, что в самом языке (стандарт ECMAScript) описаны лишь общие конструкции, которые позволяют писать программы, но не позволяют взаимодействовать с браузером.



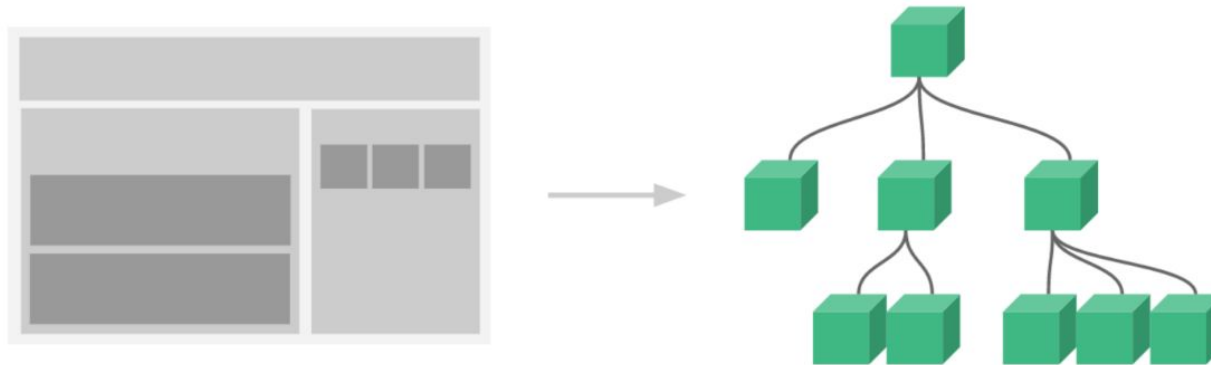
JS & Web API

Но как же тогда создаются все эти приложения? Дело в том, что браузер предоставляет скриптам (программам, написанным на JS) [Web API](#).



JS & Web API

Web API предоставляет достаточно много возможностей, но не всегда удобен, особенно для построения интерфейсов. Например, если рассмотреть типичный портал, то можно увидеть, как логически интерфейс разбивается на блоки:



Примечание: картинка из руководства Vue.js (но суть отражает правильно)

Эти блоки именуются компонентами. И хотелось бы иметь возможность создавать их независимо друг от друга, потому что, например, Новости никак не влияют на "шапку".



JS & Web API

Q: зачем их создавать отдельно, ведь можно же просто всё запрограммировать в одном файле?

A: дело в том, что когда вы создаёте большой сервис, то один и тот же блок будет повторяться на многих страницах этого сервиса.

А если он повторяется, то зачем его ещё раз программировать заново? Хотелось бы просто как лего, собирать весь интерфейс из готовых кубиков.

Эту задачу как раз и решает React.js. Но прежде чем мы перейдём к самому React'у нам надо установить ряд инструментов.



CRA



CRA

Для того, чтобы создать приложение на React, есть три ключевых пути:

1. Скачать и подключить библиотеку руками
2. Настроить сборщик проектов Webpack, установив и настроив все дополнительные инструменты
3. Использовать специальный инструмент [create-react-app](#) (CRA)

Мы с вами начнём с третьего варианта, поскольку он самый удобный и распространённый.



CRA

Для того, чтобы использовать CRA нам необходим `npm`, а точнее, инструмент `npx`, который входит в состав Node.js.

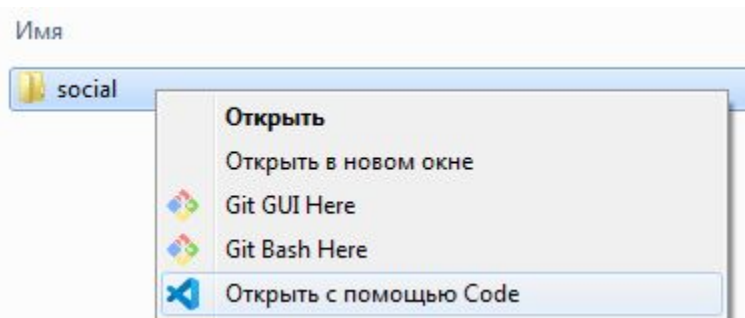


NPM



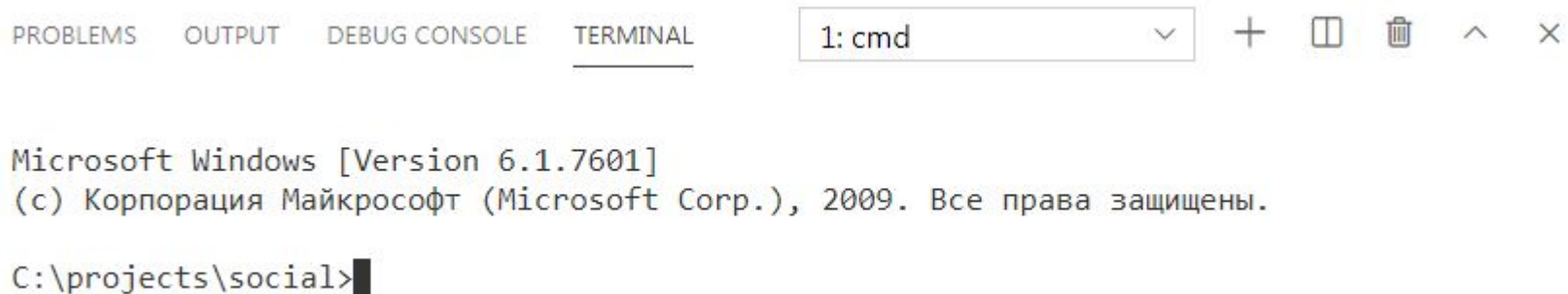
Консоль VS Code

Все команды, которые мы будем запускать далее, выполняются в консоли VS Code (или командной строке Windows). Для этого откройте каталог, в котором вы будете создавать проект (мы создали на диске `C:\projects\social`), в VS Code. Для этого нужно щёлкнуть правой кнопкой мыши на каталоге и выбрать открыть в VS Code:



Консоль VS Code

Чтобы открыть консоль, нажмите сочетание клавиш **Ctrl + `** (рядом с 1 на клавиатуре):

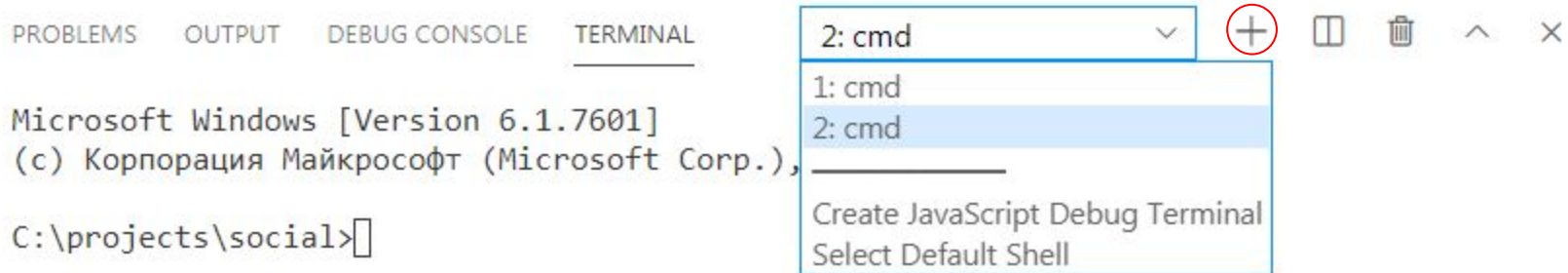


Именно здесь и нужно будет вводить команды.



Консоль VS Code

С помощью кнопки **+** вы можете создавать несколько консолей, а с помощью выпадающего списка – переключаться между ними.



npm

Итак, в состав Node.js входит специальный инструмент – [npm](#):

```
$ npm --version
```

```
6.14.8
```

```
$ npm --help
```

```
Usage: npm <command>
```

```
where <command> is one of:
```

```
access, adduser, audit, bin, bugs, c, cache, ci, cit,  
clean-install, clean-install-test, completion, config,  
create, ddp, dedupe, deprecate, dist-tag, docs, doctor,  
edit, explore, fund, get, help, help-search, hook, i, init,  
install, install-ci-test, install-test, it, link, list, ln,  
login, logout, ls, org, outdated, owner, pack, ping, prefix,  
profile, prune, publish, rb, rebuild, repo, restart, root,  
run, run-script, s, se, search, set, shrinkwrap, star,  
stars, start, stop, t, team, test, token, tst, un,  
uninstall, unpublish, unstar, up, update, v, version, view,  
whoami
```

```
npm <command> -h  quick help on <command>
```

```
npm -l            display full usage info
```

```
npm help <term>  search for help on <term>
```

```
npm help npm     involved overview
```



npm

npm – это:

- Node.js package manager (менеджер пакетов для Node.js);
- npm Registry (реестр пакетов – готовых написанных инструментов);
- npm CLI (инструмент командной строки, поставляемый вместе с Node.js).

В терминах этой системы все внешние библиотеки и инструменты называются пакетами. Эта же система хранит онлайн большой реестр пакетов (почти все – бесплатные), которые вы можете скачать и установить для собственного использования.



npm

Представляйте это себе как Google Play или AppStore – есть приложение на вашем телефоне, которое позволяет просматривать Google Play или AppStore.

С помощью него можно ставить другие приложения, которые вам нравятся, например, Telegram или WhatsApp.


При этом сами приложения Telegram и WhatsApp хранятся в этих самых Google Play или AppStore (их туда загружают сами разработчики приложений).

Т.е. это Google Play или AppStore для JS разработчиков. Мы не оговорились – именно JS, поскольку использовать React, Angular, Vue или Svelte без [npm](#) очень проблематично.



npmjs.com

Сами пакеты размещаются на сайте npmjs.com:

 Search Sign Up Sign In

135172 packages found

Sort Packages

Optimal

Popularity


Quality

Maintenance

react exact match

React is a JavaScript library for building user interfaces.


react

 **acdlite** published 16.13.1 • 5 months ago

react-dom

React package for working with the DOM.

react

 **acdlite** published 16.13.1 • 5 months ago

p

q

m

1

2

3

...

6759

»



npm

npm (как и многие другие инструменты) рассчитан на безлимитный Интернет, поэтому скачивает достаточно много файлов (по несколько сотен мегабайт на один проект).

Поэтому будьте аккуратны при его использовании, выбирайте один из вариантов:

1. Копируйте каталог с предыдущим проектом (вместе с `node_modules`)
2. Используйте Git



CRA



Инициализация пакета

Мы находимся в каталоге `C:\projects\social` и в нём будем создавать пакет с помощью CRA.

CRA – это такой конструктор вашего приложения, который за вас сделает первоначальные настройки, скачает React и создаст скелет приложения.



Инициализация пакета

В консоли запустите следующую команду: `npx create-react-app .` (пробел и точка обязательны):

```
Success! Created social at C:\projects\social  
Inside that directory, you can run several commands:
```

```
npm start
```

```
Starts the development server.
```

```
npm run build
```

```
Bundles the app into static files for production.
```

```
npm test
```

```
Starts the test runner.
```

```
npm run eject
```

```
Removes this tool and copies build dependencies, configuration files  
and scripts into the app directory. If you do this, you can't go back!
```

We suggest that you begin by typing:

```
cd C:\projects\social
```

```
npm start
```

Happy hacking!

```
C:\projects\social>█
```



Инициализация пакета

Важно: установка будет не быстрой, поэтому придётся подождать.



Запуск сервера разработки

Теперь нам нужно запустить сервер разработки (он автоматически будет отслеживать все изменения и обновлять веб-страницу в нашем браузере):

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

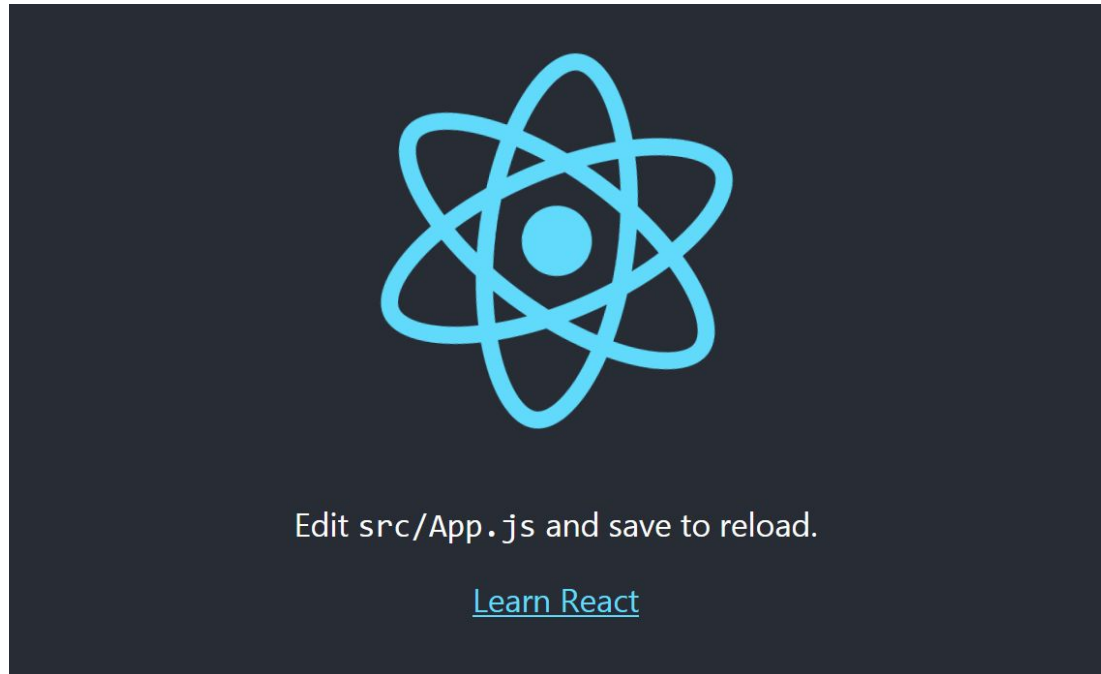


```
C:\projects\social>npm start  
  
> social@0.1.0 start C:\projects\social  
> react-scripts start  
  
Starting the development server...
```



Запуск сервера разработки

Если вы сделали всё правильно, то у вас откроется веб-браузер со следующей страницей:



Остановка сервера разработки

Чтобы прекратить работу сервера разработки нажмите сочетание клавиш **Ctrl + C**.



Сборка

Теперь давайте попробуем собрать наше приложение (чтобы получить файлы, которые и будут публиковаться на сервере, например, на GitHub Pages). Для этого выполните команду:

`npm run build` (можно в соседней вкладке терминала)

Важно: некоторые команды npm можно выполнять без `run` (например, `npm start`, `npm test`), для остальных же нужно добавлять `run` (`npm run build`). Дело в том, что так реализована поддержка скриптов ([скрипты](#) – это специальная секция с командами в вашем файле `package.json`: туда записывают стандартные для текущего проекта команды).



Сборка

После сборки у вас появится каталог `build` ("числа" рядом с именами файлов у вас могут быть другими):

```
$ npm run build
```

```
> social@0.1.0 build
> react-scripts build
```

```
Creating an optimized production build...
Compiled successfully.
```

File sizes after gzip:

```
46.61 kB  build/static/js/main.54ce199b.js
1.78 kB   build/static/js/787.9181411e.chunk.js
541 B     build/static/css/main.073c9b0a.css
```

The project was built assuming it is hosted at `/`.
You can control this with the `homepage` field in your `package.json`.

The `build` folder is ready to be deployed.
You may serve it with a static server:

```
npm install -g serve
serve -s build
```

Find out more about deployment here:

<https://cra.link/deployment>



Сборка

Посмотреть, как полученная нами сборка будет работать вживую в браузере можно с помощью команды `npx serve build` (т.е. мы запустим `serve` для демонстрации содержимого нашего каталога `build`):

```
<!DOCTYPE html>
<html lang="en">
  <head>...</head>
  <body> == $0
    <noscript>You need to enable JavaScript to run this app.</noscript>
    <div id="root">
      <div class="App">
        <header class="App-header"> flex
          
          <p>
            "Edit "
            <code>src/App.js</code>
            " and save to reload."
          </p>
          <a class="App-link" href="https://reactjs.org" target="_blank" rel="noopener noreferrer">Learn React</a>
        </header>
      </div>
    </div>
  </body>
</html>
```

Сборка

Содержимое DOM-дерева в режиме разработки и в режиме "production" сборки будет немного отличаться.

Полученный нами каталог уже можно использовать для публикации на сервере (он заточен на то, чтобы располагаться в корне веб-сайта, например, <https://alif-skills.pro/>).

Если же вы собираетесь публиковать не в корне сайта, а например, в подкаталоге (так делает GitHub Pages) – <https://alif-skills.github.io/react/>, то в `package.json` нужно [добавить соответствующие настройки](#).



ИТОГИ



Итоги

В этой лекции мы обсудили достаточно много важных моментов:

1. Установку инструментов
2. Запуск нашего приложения

В следующих лекциях мы будем опираться на то, что вы уже изучили в этой лекции (и не будем детально описывать процессы инициализации проекта и т.д.).



ДОМАШНЕЕ ЗАДАНИЕ



Орг.моменты

Практикум состоит из 8 обязательных занятий. Мы выкладываем новые занятия каждый понедельник в 14:00 (по Душанбе).

Каждое воскресенье в 23:59 (по Душанбе) дедлайн сдачи домашнего задания. Дедлайн – это предельный срок, до которого вы должны сдать ДЗ.

Если не успеете сдать в срок домашнее задание, тогда этот практикум будет для вас закончен и вы сможете зарегистрироваться на запуск следующего через несколько месяцев.

Все вопросы вы сможете задавать в [Телеграм канале](#).



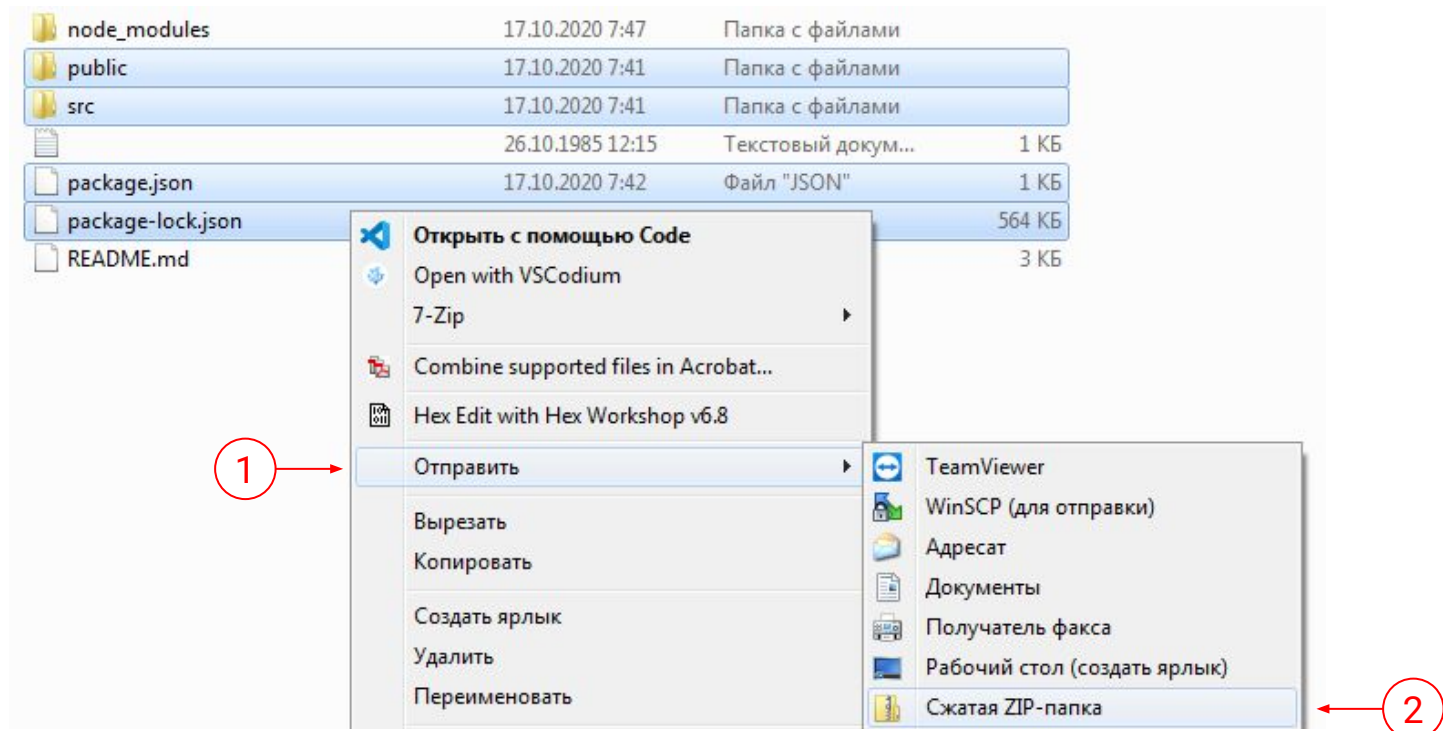
ДЗ: Hello React!

Создайте проект аналогично тому, как мы это делали на лекции. Отправьте файлы, так, как указано на скриншоте на следующей странице.



Как сдавать ДЗ

Вам нужно запаковать в zip-архив ваш проект те файлы и каталоги, которые указаны на скриншоте ниже. Для этого выберите их, нажмите правую кнопку мыши и выберите **Отправить (1) Сжатая ZIP-папка (2)**:

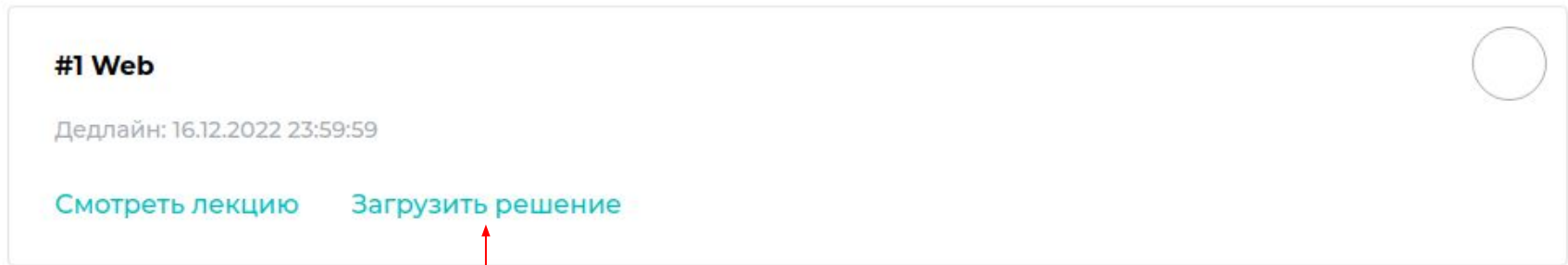


Важно: `node_modules` архивировать не нужно!



Как сдавать ДЗ

Перейдите в личный кабинет на сайте <https://alif-skills.pro> и выберите **Загрузить решение:**



Как сдавать ДЗ

На открывшейся странице выберите нужную задачу и нажмите [Загрузить решение](#) (вам предложат выбрать ваш zip-файл):

Решения

Hello, HTML!

Баллов: 1 Обязательна: Да Дедлайн: 16.12.2022 23:59:59

 [Загрузить решение](#)

Только файлы формата zip, размером не более 1 мегабайта.

Важно: учитывается **только последняя попытка**.

Нет попыток

[Назад](#)



Неверное решение

Если вы загрузите неверное решение, то в поле статус будет написано **НЕ ПРИНЯТО** (1) и вам нужно будет загрузить исправленное решение (2):

② → Загрузить решение

Только файлы формата zip, размером не более 1 мегабайта.

Важно: учитывается **только последняя попытка**.

✓ Последняя попытка

①

#7 Статус: **НЕ ПРИНЯТО** Отправлено: 10.12.2022 00:58:47 Проверено: 10.12.2022 00:58:56

```
START
Распаковываем архив - OK
Проверяем каталог docs - FAIL
В вашем архиве не найден каталог docs
docs directory not found

FAIL
Проверки завершились с ошибкой.
```

Количество попыток не ограничено.



Как читать ошибки

Открываете результаты и идёте сверху вниз (1) до первой записи **FAIL** (2):

Загрузить решение

Только файлы формата zip, размером не более 1 мегабайта.

Важно: учитывается **только последняя попытка**.

✓ Последняя попытка

```
#7 Статус: НЕ ПРИНЯТО Отправлено: 10.12.2022 00:58:47 Проверено: 10.12.2022 00:58:56

1 START
  Распаковываем архив - OK
  Проверяем каталог docs - FAIL ← 2
  В вашем архиве не найден каталог docs
  docs directory not found

  FAIL
  Проверки завершились с ошибкой.
```

Ниже будет написана причина ошибка – "В вашем архиве не найден каталог docs", т.е. мы забыли в архив положить папку docs или назвали её неправильно.



Как читать ошибки

Если ошибка будет в структуре вашего HTML, то:

- + как должно быть
- как у вас

```
START
Распаковываем архив - OK
Проверяем каталог docs - OK
Проверяем файл index.html (в каталоге проекта) - OK
Проверяем приложение
Пишем тест - OK
Запускаем тесты - FAIL
FAIL ./test.js
  × Проверка документа (238 ms)

    • Проверка документа

      + Версия бота
      - Ваша версия

        <body>
      +   <h1>Hello, HTML!</h1>
      -   <h1>Hello, Web!</h1>

        </body>
```



Как читать ошибки

В некоторых случаях бот будет от вас требовать определения атрибутов в определённой последовательности – обращайтесь на это внимание!



Успешная сдача

После того, как вы успешно выполните все требования задания, оно будет отмечено как зачтённое (пересдать его будет нельзя):

Hello, HTML!



Баллов: 1 Обязательна: Да Дедлайн: 16.12.2022 23:59:59

Задача сдана

✓ Последняя попытка

#30 Статус: **ПРИНЯТО** Отправлено: 10.12.2022 04:06:47 Проверено: 10.12.2022 04:06:51

```
START
Распаковываем архив - OK
Проверяем каталог docs - OK
Проверяем файл index.html (в каталоге проекта) - OK
Проверяем приложение
Пишем тест - OK
Запускаем тесты - OK
OK
```



Успешная сдача

Только после сдачи всех обязательных задач к текущему занятию вам будет доступны материалы следующего занятия.



Спасибо за внимание

alif skills

2023г.

